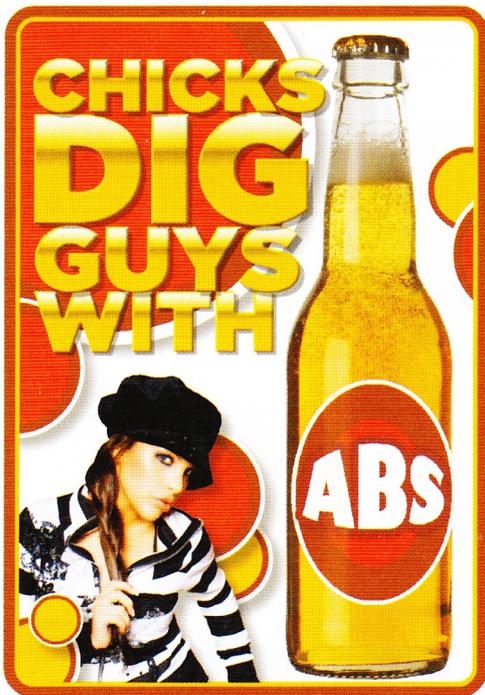
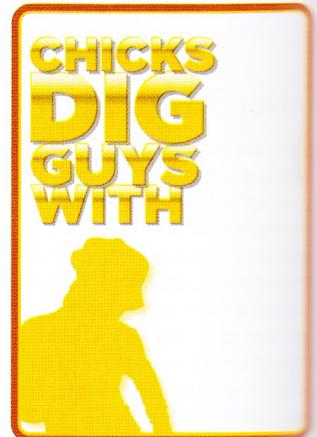


Pausing the Timeline

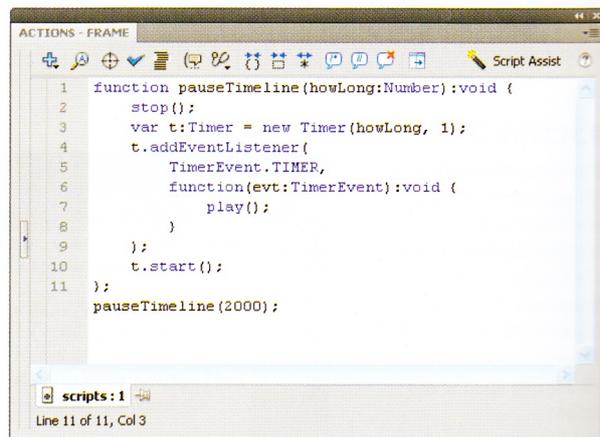


SOONER OR LATER you're going to want to pause the Timeline for a few seconds or even a few minutes, then resume the action after the viewer has had the chance to read something important or cast a lengthy gaze at your artwork. Flash banner advertisements are commonplace throughout the Web and the ability to pause the playhead to help emphasize a specific advertisement message can be critical to the success of the ad itself.

The obvious way to handle this is to add extra frames where you want the pausing to occur. Depending on the length of the pause, though, that can make for an unwieldy Timeline, especially if you have a high frame rate (five seconds at 30 fps is 150 frames of "dead space"). What if you could pull that off with a single frame?



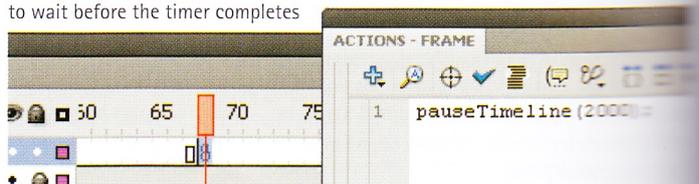
ActionScript 3.0

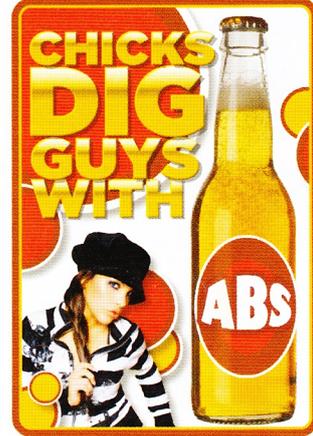
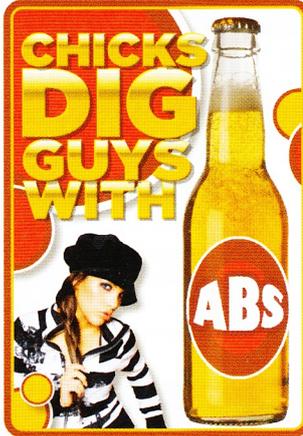
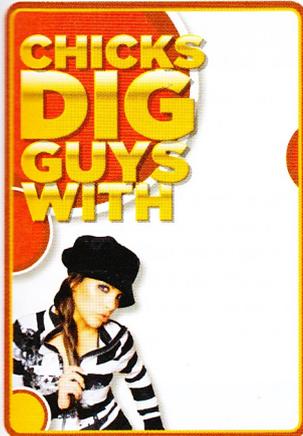


1 This pause mechanism comes in two parts. If you only need to pause once, put both parts in the same frame; otherwise, put the function in frame 1, then call that function in later frames as needed. This is like an "everything" burrito. Everything you need is stuffed into this custom function. The first thing it does is stop the Timeline. Next, it declares a variable, *t*, and sets it to an instance of the *Timer* class. The two parameters, *howLong* and *1*, tell Flash how long to wait before the timer completes

and how often to run the timer, respectively. Once the *Timer* variable is in place, things start looking familiar again: a *TimerEvent.TIMER* event is added to the *t* instance and told to perform a function that causes the Timeline to play again. Finally, the timer is started.

Once the function is in place, call it by name from the same frame or any subsequent frame. The value *2000* refers to 2000 milliseconds (that is, two seconds).





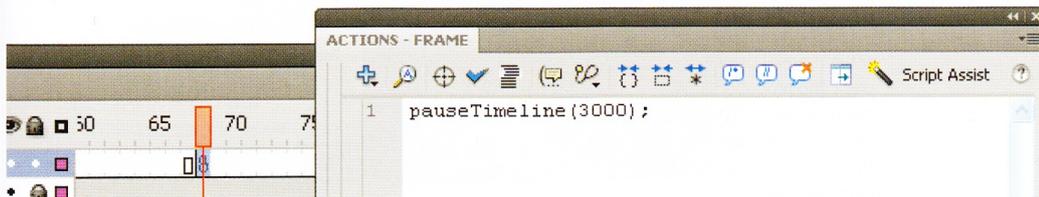
ActionScript 2.0

```

ACTIONS - FRAME
Script Assist
1 function pauseTimeline (howLong:Number):Void {
2     stop();
3     var id:Number = setInterval(
4         function():Void {
5             play();
6             clearInterval(id);
7         },
8         howLong
9     );
10 };
11
    
```

2 The AS2 version relies on the `setInterval()` function, which repeatedly triggers whatever other function you tell it to, as provided in its first parameter. The second parameter here, `howLong` defines when the triggering should occur, in milliseconds. A companion function, `clearInterval()`, stops the repeated triggering after a single run. How? Earlier, `setInterval()` returns a `Number` value that gets stored in the `id` variable, then `clearInterval()` references that number by way of `id` as its own parameter.

Here, again, you've got two parts. Put the first in frame 1. Then call it in that same frame or any later frame(s). Calling the function is the same as in the AS3 version.



HOT TIP

Timers are useful for all sorts of purposes, and you may not always want your timer to stop. In AS3, specify 0 as the second parameter in new `Timer()` to loop forever. In AS2, don't call `clearInterval()`.